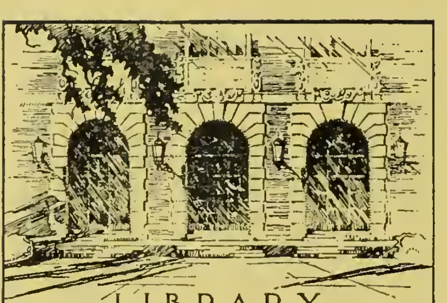


[redacted]

DIGITAL COMPUTER LABORATORY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

151

REPORT NO. 148

THE ILLINOIS PATTERN RECOGNITION COMPUTER
(ILLIAC III)

by

Bruce H. McCormick

(Presented at the
18th Annual ACM National Conference,
Denver, Colorado
August 27, 1963)

August 20, 1963

This work was supported in part by
Contract No. AT(11-1)-1018 of the Atomic Energy Commission

ACKNOWLEDGMENT

Five of my colleagues on the faculty of the Digital Computer Laboratory have made major contributions to the design and fabrication of ILLIAC III: Mr. Kimio Ibuki (taxicrinic unit), Professor R. Narasimhan (programming and the syntactic model), Professor Sylvian R. Ray (transfer memory), Professor K. C. Smith (circuit design of the pattern articulation unit and computer power distribution), and Professor Shigeharu Yamada (fast circuits and the automatic test console). A selected bibliography will be found appendaged to this paper where the original reports of these associates are noted.

In addition I would like to acknowledge the contribution of former graduate students, particularly those who weathered the days when neither the structure nor the feasibility of the computer was very certain: J. L. Divilbiss (the bubble register), Dennis Hall (stalactite circuitry), James H. Stein, Brian Mayoh and R. Kevin Rice (simulation programming), Cyril P. Bates (scanner digital control), and S. Matsushita (arithmetic unit).

The structure of the Pattern Articulation Unit (PAU) is an outgrowth of the recent interest in cellular automata, in particular the work of S. H. Ungar.[1] Professor A. H. Amon was instrumental in the importance given to continuity in the machine--which developed then into the flash-through mechanism.

I wish to thank Mrs. Phyllis Olson for typing the manuscript and both James Burrell and Paul Richardson for preparing the illustrations.

In addition I would like to acknowledge the very considerable assistance of Joseph V. Wenta, who has been responsible for supervising the checkout operation and maintenance of our prototype flying spot oscilloscopic system and the construction of the second generation device. In like manner Frank P. Serio has ably supervised the fabrication of the main frame and power distribution system of the computer.

---

[1] S. H. Unger, "A Computer Oriented Toward Spatial Problems," Proc. IRE, Vol. 46, pp. 1744-1750 (1958); and "Pattern Detection and Recognition," Proc. IRE, Vol. 47, pp. 1737-1752 (1959)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

The Illinois Pattern Recognition Computer (ILLIAC III), an all-digital processor for visual information, is under construction at the Digital Computer Laboratory, University of Illinois. This visual recognition digital computer (Fig. 1) is designed for automatic scanning and analysis of massive amounts of relatively homogeneous visual data. In particular the design is an outgrowth of studies at this Laboratory of a computer system capable of scanning, measuring, and analyzing in excess of 10^7 bubble chamber negatives* per year.[8,9,10,11,12,13]

The computer, when complete, will comprise:

(1) High-speed oscilloscopic scanners for the rapid ingestion into the visual recognition computer of photographic information digitized on a black-white raster. These devices also have a measuring mode capable of obtaining accurate coordinate digitization of points lying on tracks or other entities appearing in the picture.

(2) A Pattern Articulation Unit (PAU), whose duty is to perform local preprocessing on the digitized raster, such as track thinning, gap filling, line element recognition, etc. The logical design of the all-digital processor has been optimized for the idealization of the input image to a line drawing. Nodes representing end points, bends, points of intersection are labeled in parallel by appropriate programming. The abstract graph describing the interconnection of labeled nodes is then extracted as a list structure, which comprises the normal output of the processing unit.

(3) A Taxicrinic Unit (TU),[19] which assembles such graphs into coherent list structures subject to a manipulative grammar, and categorizes these graphs to complete the visual recognition function.

---

* Typically a stereo-exposure of a bubble chamber comprises three negatives.

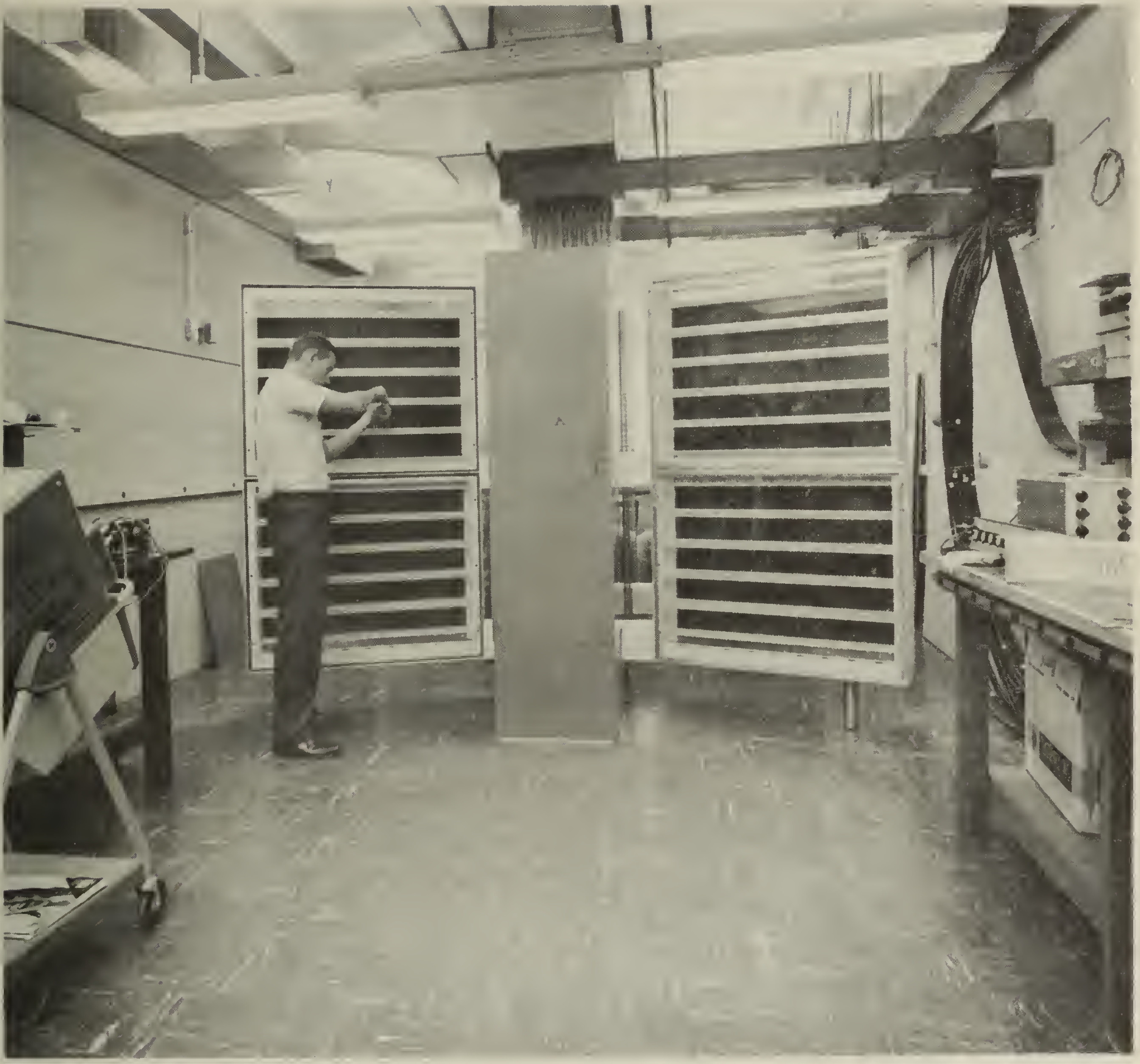


Figure 1. Main Frame of the Computer (as of June 1963)

(4) An Arithmetic Unit[20] for executing the mathematical analysis (stereo-reconstruction, statistical summary, etc.) involved in processing photographic data.

(5) An Output Complex consisting of printers, visual displays, etc., for presenting and storing the output of the system.

Every attention is being given to design and construct the system so that partial but increasing production capability is achieved at each earlier stage in the complete development, with simulation of uncompleted units provided by the attached ILLIAC II computer of the Digital Computer Laboratory.

This paper will principally discuss the use, design and fabrication of the Pattern Articulation Unit (PAU), as this processor of the computer is of fundamentally new design and therefore least familiar. The PAU employs a two-dimensional iterative array of 1024 (32 x 32) identical processing modules locally connected to execute boolean functions, threshold logic and signal path building. It is augmented by an unconventional core memory, called a transfer memory, which in conjunction with the iterative array, can operate as an associative memory.

2. MODE OF VISUAL ANALYSIS

2.1 Scanning and Measuring

Visual material must be digitized before input to the pattern recognition computer proper. The input mechanism which performs this task will be referred to as the scanning unit. Input visual information is typically in frames: one frame per view of a bubble-chamber stereo-triad, field-of-view of a neurohistological section, or microfilm frame of printed text (Fig. 2).[1]

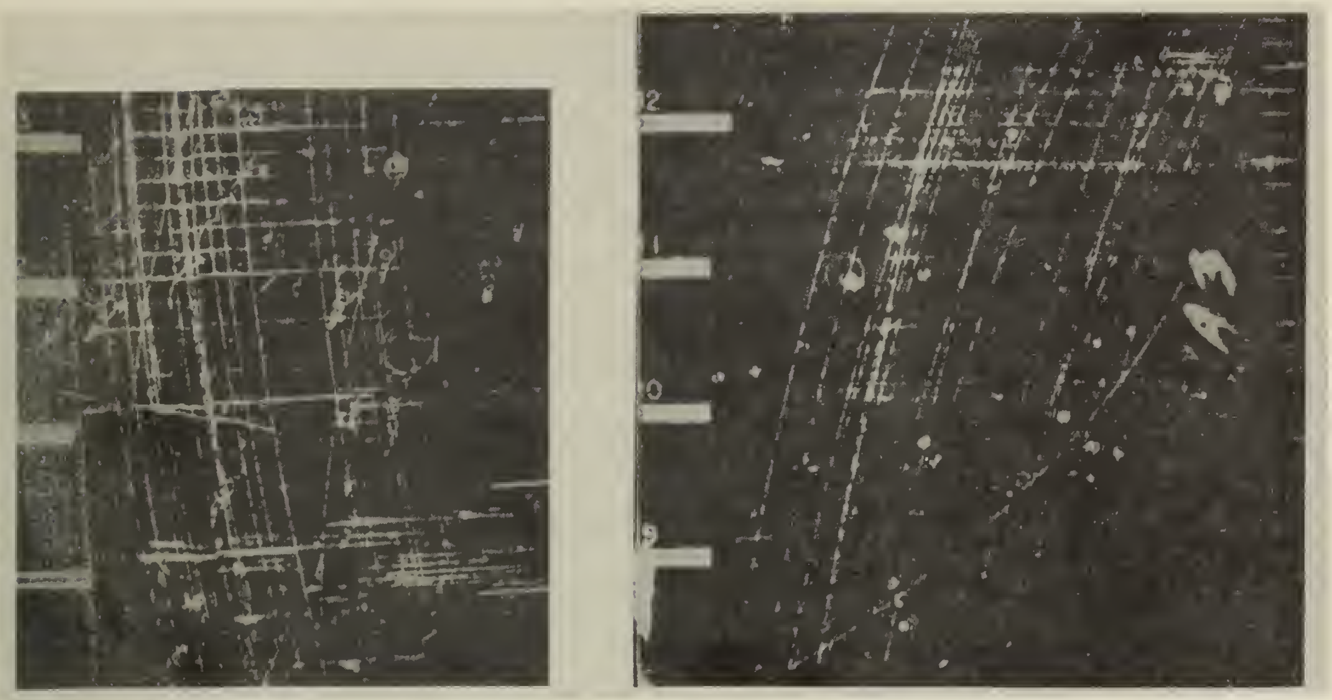
All scanning units currently under construction at this Laboratory examine the film (or biological section, etc.) by a TV-like scan; that is, the input image has imposed upon it a two-dimensional rectangular (or rhombic) raster. Digital representation of the input material is normally one of two dominant forms: (1) bit, or (2) coordinate.

For example, one view of a bubble-chamber stereo-triad can be digitally encoded in bit form by projecting a raster (1024 x 4096 cells) upon the frame. Each cell is examined to see if the negative locally is above some minimum opacity. If so, a "1" is recorded; alternatively, a "0" is recorded, indicating that the negative is locally transparent (i.e., background only). Typically, for the bubble-chamber case, encoded tracks range from one to three cell lengths in width.

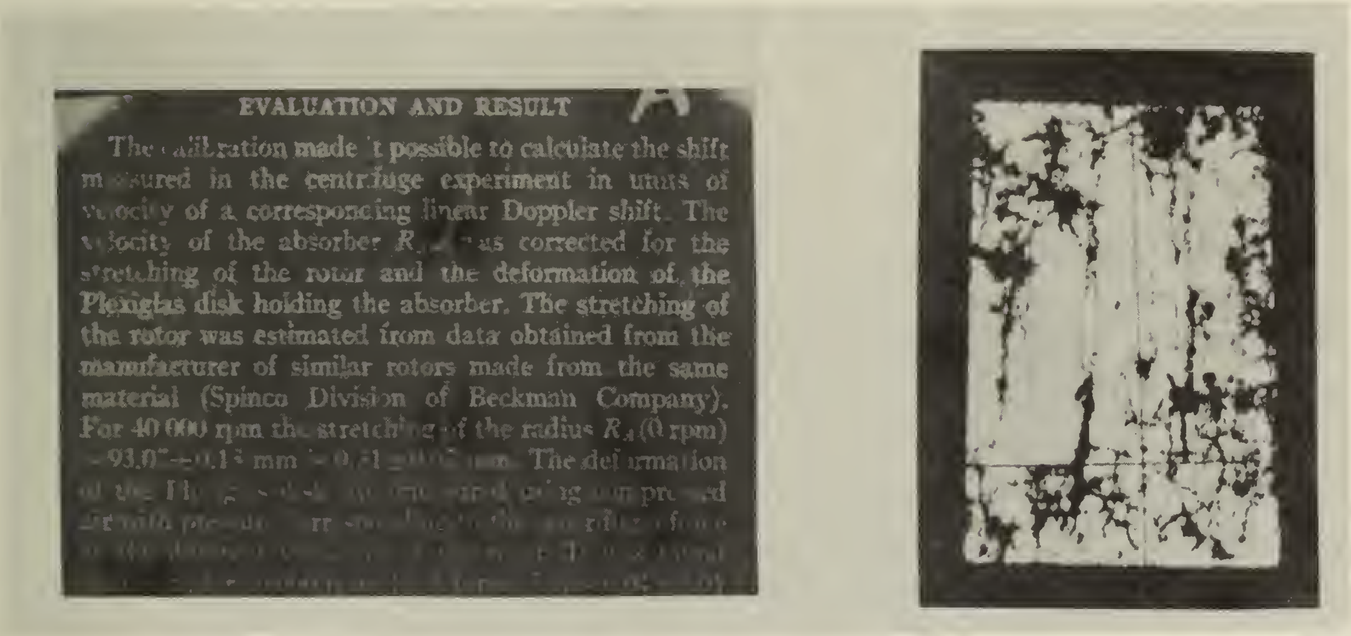
Alternatively the view can be digitally encoded in coordinate form by scanning the frame, again for the bubble-chamber case, with 4096 parallel sweeps (parallel to the X axis) and for each sweep recording the X coordinate for each leading (and trailing) edge of every black mass seen. In bubble-chamber case the total number of bits required is approximately equal for the two digital representations.

In the digitizing process it is convenient to distinguish between scanning and measuring. For some areas, e.g., alpha-numeric recognition, measuring normally plays no role. In general, measuring plays an ancillary role to scanning.

The first image filtering, or recognition processes, would appear nominally independent of mild image distortion. In fact a human scanner can find, without difficulty, events of interest in playback photographs of bit-encoded film even though the oscilloscopic scan/playback system has introduced distortions of five to ten percent (compare Fig. 2).



(a) Bubble-Chamber Negatives



EVALUATION AND RESULT

The calibration made it possible to calculate the shift measured in the centrifuge experiment in units of velocity of a corresponding linear Doppler shift. The velocity of the absorber R_A was corrected for the stretching of the rotor and the deformation of the Plexiglas disk holding the absorber. The stretching of the rotor was estimated from data obtained from the manufacturer of similar rotors made from the same material (Spinco Division of Beckman Company). For 40 000 rpm the stretching of the radius R_A (0 rpm) [illegible] The deformation [illegible]

(b) Scientific Text

(c) Stained Rat-Brain Tissue

Figure 2. Scope Playback of Flying Spot Scanned Material (1024 Line Scan)

In contrast, for measurement, resolution of one part in ten of a track width (2μ on the film) is employed and comparable linearity held over the entire face of the film. This added coordinate resolution is required only for (typically) ten percent of the tracks in a frame--these slower-speed high-resolution procedures can be called into play where needed, without significant deterioration of the over-all processing rate. Accordingly a traditional separation of visual data processing into a scanning and a precision measuring phase has grown up.

It is entirely consistent to use one instrument for both scan and measurement purposes. The second-generation oscilloscopic systems[14] under development at this Laboratory give promise of handling both functions: a very fast (open-loop) scanning mode applied universally; a slower (closed-loop, i.e., spot referenced to an optical grating) measuring mode of those film areas singled out as a result of processed scan information.

Neither the scanning unit, nor the measuring unit, performs a pattern recognition function. Input bit patterns are arbitrary, as far as these units are concerned, and reflect the general-pupose nature of the over-all recognition system. Rather, the interpretation of the input bit information (i.e., visual recognition) is a function of the subsequent processor--the visual recognition computer proper.

2.2 Pattern Articulation

A scanning program set up, for example, to process bubble-chamber negatives examines one view at a time.[8] Each negative is partitioned into a network of windows, a window representing a square raster of size 32 x 32 bits. For the particular case of exposures made of the 72-inch hydrogen bubble chamber made at 15:1 demagnification, it is our experience[9] that a partition of the picture into roughly 15 windows across and 60 windows in length provides marginally adequate resolution--approximations 10^3 windows per view. These windows are then processed sequentially, in a TV-like scan, starting with the bottom left corner window and ending with the top right window.

In this procedure all in-window processing is executed by the Pattern Articulation Unit (PAU); cross-window compilation of accumulated information, however, is delegated to the Taxicrinic Unit (TU).

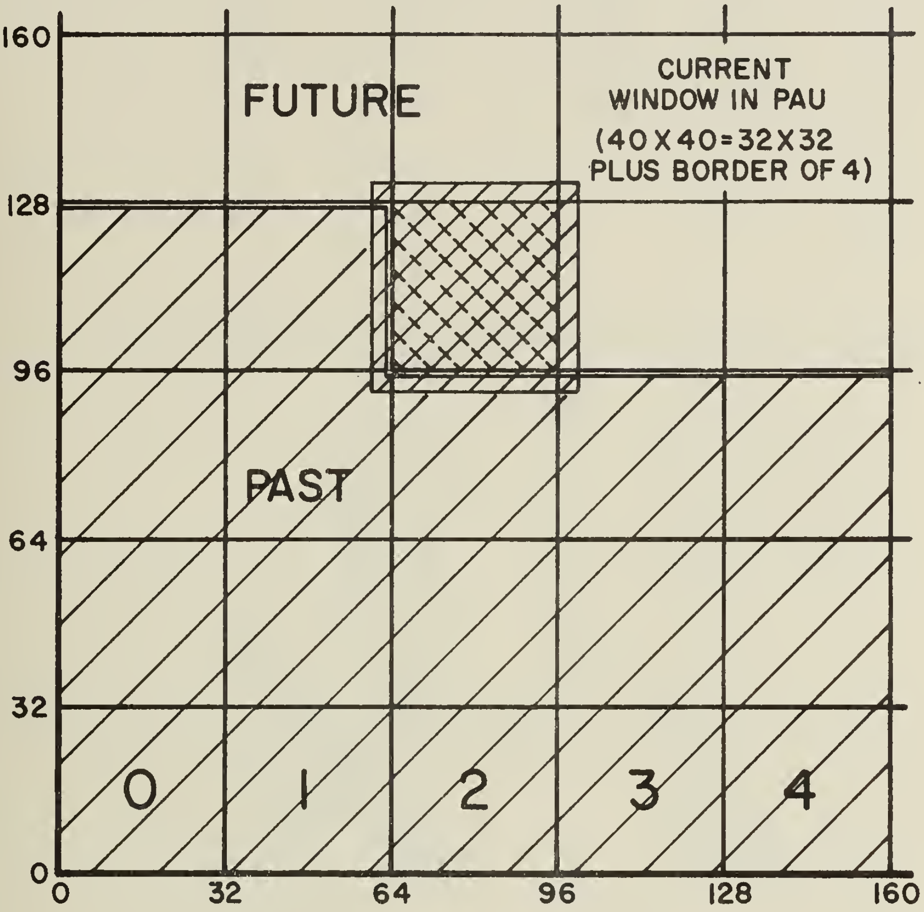
Within each window we require an iterative procedure (1) to update any track segments previously named which enter the window; (2) to identify vertices and name new tracks which originate within that window. This boundary value information, and only this informaion, is then transmitted to the Taxicrinic Unit. Accordingly the input-output "black box" functional description of the PAU is specified by the input (a window) and this boundary value output (Fig. 3).

The image within each window must be idealized by edging, line thinning, gap filling and line smoothing algorithms. These transformations, dependent only upon local behavior of the image, can be performed in parallel on the image. When correctly selected, they idealize the original image to a line drawing (Fig. 4).

This line drawing can be converted to a linear graph by appropriate labeling (in parallel) of vertices, bends, crossovers, and terminals of the idealized image (Fig. 5). We have chosen to call the processor that performs these two functions--idealizing and labeling--the Pattern Articulation Unit (PAU), in line with the root meaning of the verb articulate: to utter distinctly, properly; to divide into joints. This labeled line drawing, or graph, is still embedded in the processing array, and procedures will now be introduced for its extraction and description.

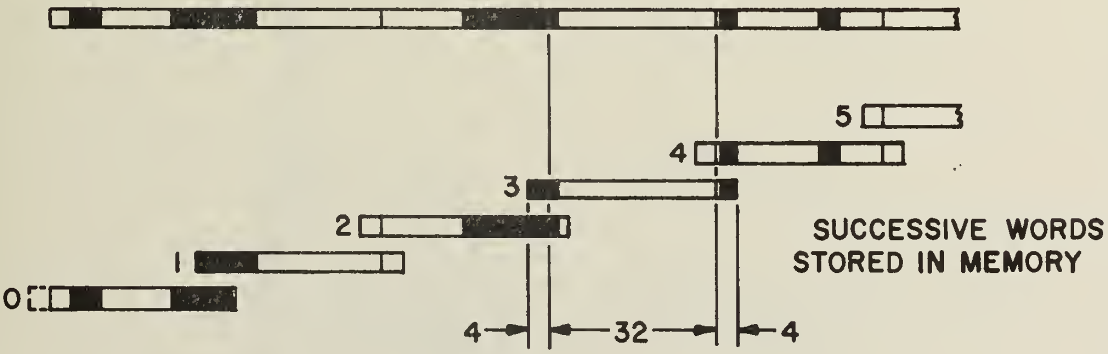
A natural description of the graph is to list in turn each edge, or branch. Each branch (connected line segment) is described by a pair of terminal points, or nodes, augmented on occasion by a tag: a few bits of qualitative information (crude orientation, thickness, etc.). Description of a line drawing by its labeled nodes implies a considerable data condensation. For example a 32 x 32 raster containing three nonintersecting tracks requires 1024 bits to describe as a retinal matrix, 60 bits to list as three branches (i.e., pairs of 10-bit coordinates).

This loss of information, however, reflects in subsequent ambiguity about the curve spanning given end points. A straight line segment between two nodes of the branch can be taken as a model of the actual arc, which can be considered in Figs. 6a and 6b to be in satisfactory agreement, but very inadequate in Figs. 6c and 6d. In general, continuity established between two nodes is, of itself, insufficient information. It is for this reason that labeling procedures that mark nodes (in parallel in the PAU), not only at end points, but also at



WINDOW BY WINDOW SCAN OF PICTURE BY PAU

FIGURE 3B



STORAGE FORMAT FOR THE FLYING SPOT TV SCAN

FIGURE 3A

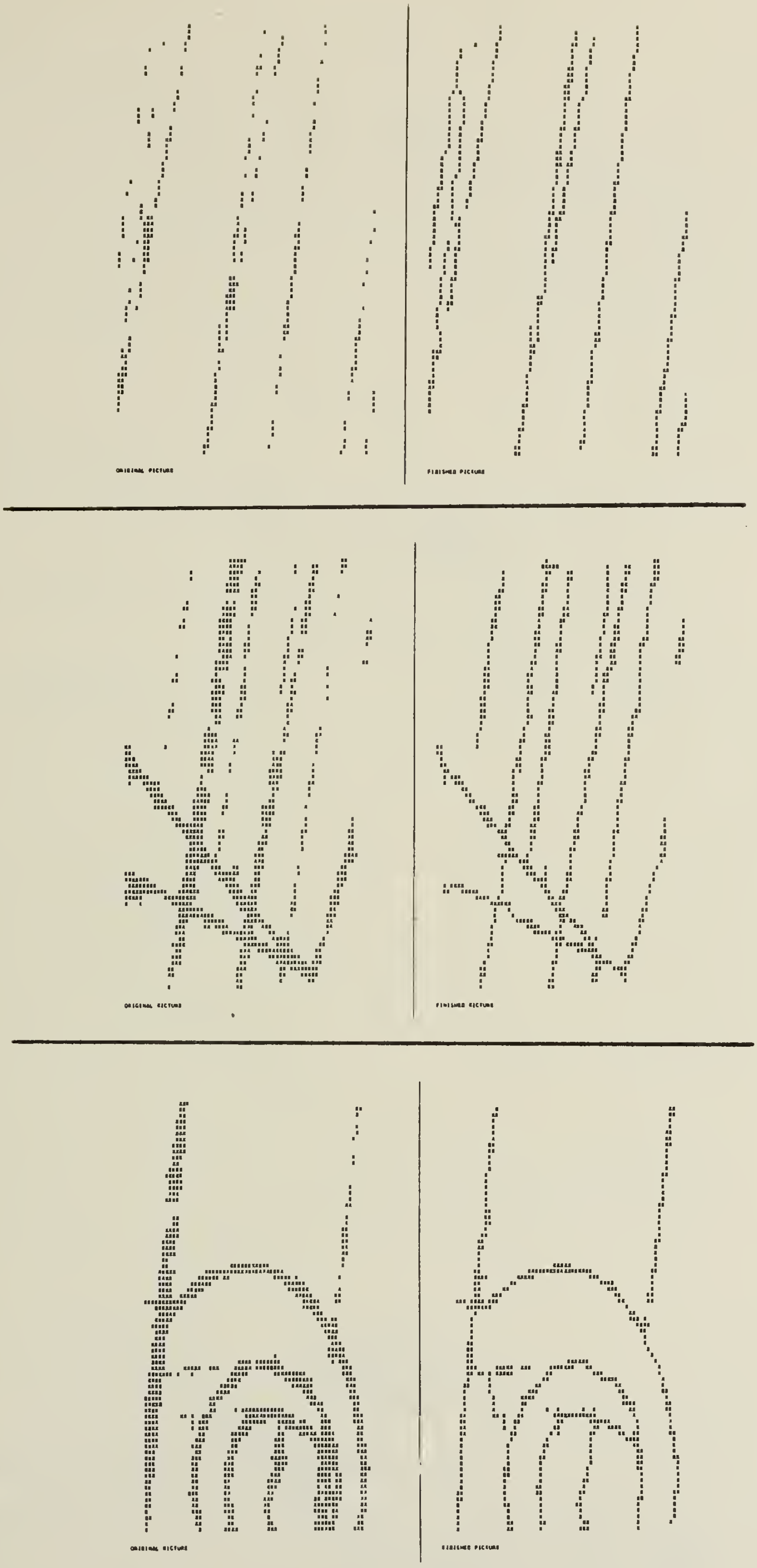


Figure 4. Image Idealized to a Line Drawing

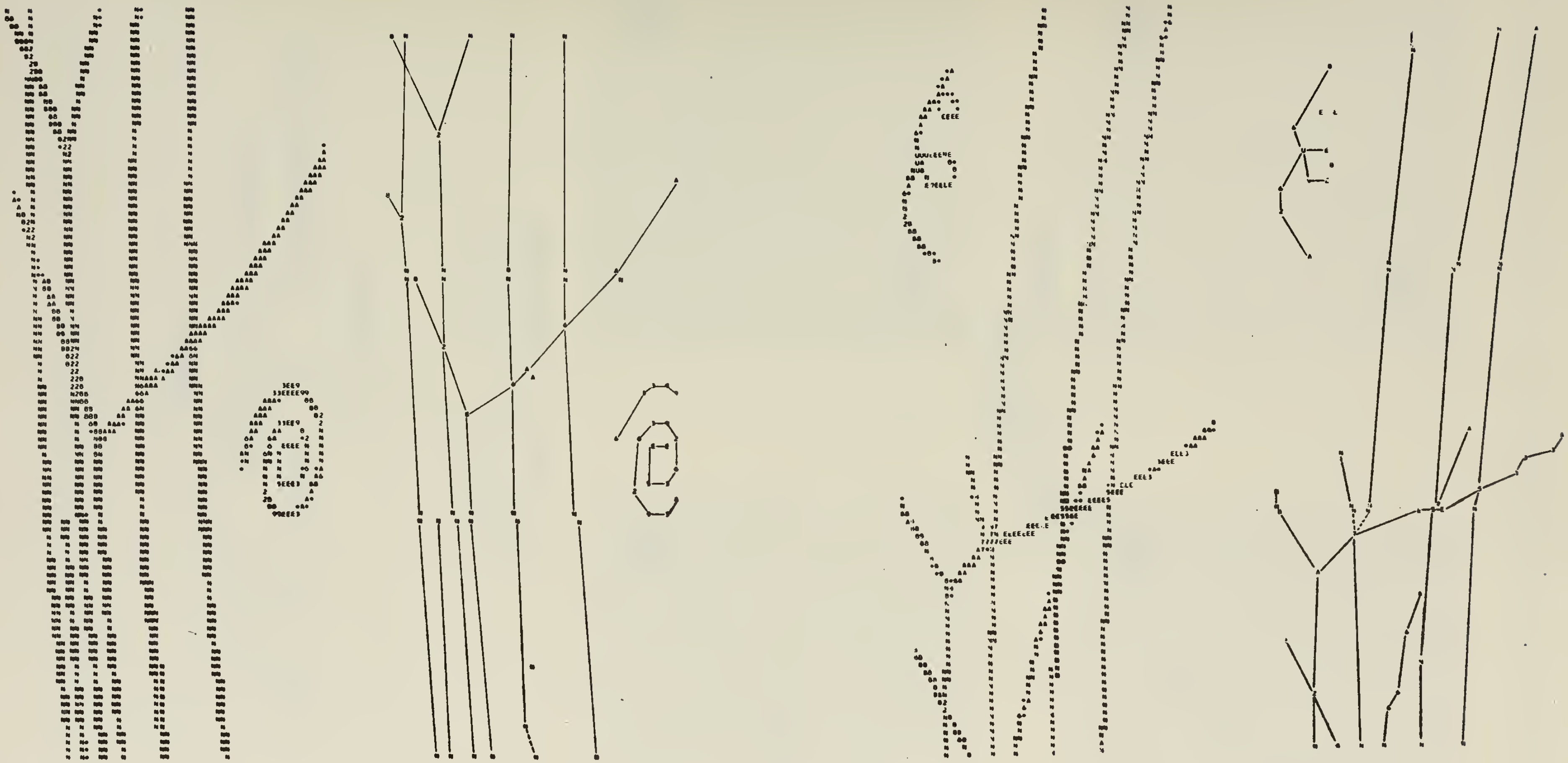
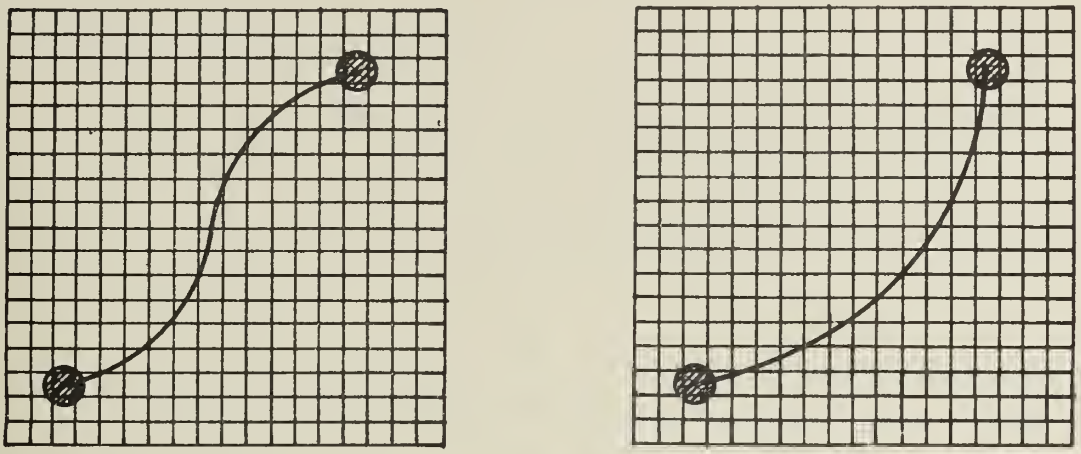
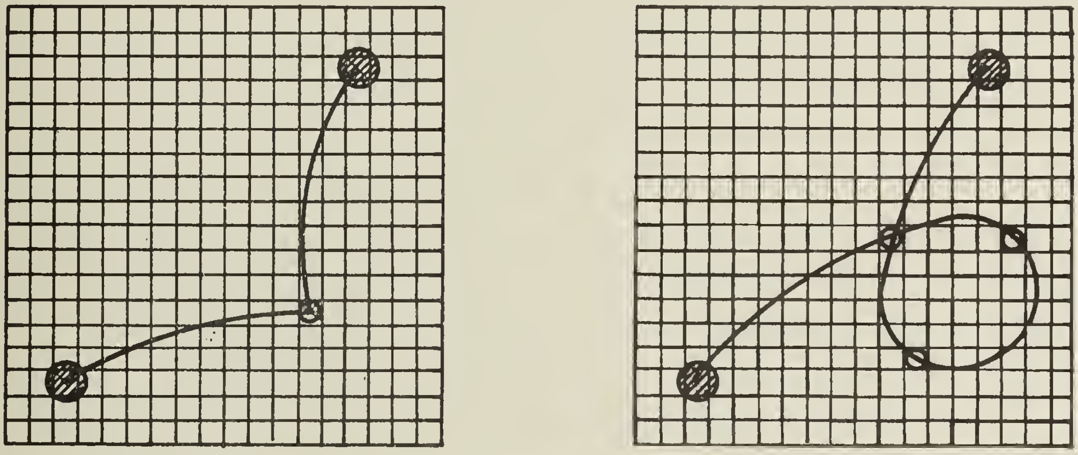


Figure 5. Labeled Bubble-Chamber Negatives



TERMINAL LABELING SUFFICIENT



ADDITIONAL INTERIOR LABELING REQUIRED

FIGURE 6

REPRESENTATION OF CURVE BY END POINTS & ADDITIONAL LABELING NODES.

junction points and positions of high curvature,, have been introduced. Introduction of open-circle nodes in Fig. 6c and 6d allow one again to fall back upon the linear graph approximation.

To mechanize a description procedure for these linear graphs it is necessary (1) to provide a means to read out the coordinates (X,Y) of all nodes in the labeled image, and (2) to give a pairing algorithm for listing together nodes sharing a common branch. Coordinate readout will be treated later, the pairing algorithm here.

A pairing algorithm reduces by definition to a tracking procedure to walk from one given node to one (or more) other terminal nodes linked to it by a continuous path of black points. This tracking procedure is inherently iterative and serial, and would appear ill-adapted to parallel processing. However, the following strategy is employed in the PAU: We build a path (in parallel) between terminal nodes. Each site position of the 1024-bit window raster is provided with a selector switch, preset in parallel by judicious programming, to link each cell to each (say) black cell of its eight immediate neighbors. If now a logical "1" signal is impressed at one terminal, the entire path rapidly assumes this signal level (flash-through), so identifying points common to this chain. In particular, all nodes connected to the initiating node are selected out. Tests of circuit design indicate that the delay per site of the propagated tracking signal can be held to approximately 60 nanoseconds per site, or typical times to identify an associated node of approximately 2 μsec.

2.3 Bilateral List Processing

At the conclusion of the pattern articulation process the original image had been reduced, window by window, to an idealized line drawing. This line drawing, in turn, was labeled and further abstracted to a graph--that is, a set of nodes and branches (pairs of nodes sharing a simple line element). Nodal coordinates were then serially extract generating a list structure.

That portion of the list structure extracted from any one window, however, may reflect ideosyncracies of local order within that one segment of the total frame and only poorly reflect long-range (global) order of the picture. That is, the processor for pattern articulation can in no satisfactory way anticipate global order--and correctly so, for the data rate lies predominantly at the local recognition level. It is, then, necessary to take the elements

found in successive windows and rearrange these sublists into descriptive categories of common speech: the tracks, electron spirals, etc., of bubble-chamber exposures or the alpha-numeric characters of printed text.

In the Pattern Recognition Computer these tasks are delegated to a processor we shall refer to as the Taxicrinic Unit (TU)* to emphasize its principal activity: the manipulation, search, and systemization of abstract graphs (bilateral list structures). These processes can, of course, be carried on within a contemporary arithmetic computer, and programming techniques such as list processing have been introduced to expedite the writing of code here. It is increasingly recognized, however, that the contemporary arithmetic computer is poorly matched to analysis situations making virtually no use of arithmetic, in which the most commonly employed order is the transfer instruction. Processors specifically adapted to graph manipulation and interpretation can be expected within the next few years to take their place in importance and utility side by side with the arithmetic unit.

2.4 Syntactic Model of Visual Description

In summary a model--called the Syntactic Model--for the description of visual data has been implicitly proposed here.[5,6,7] In brief this descriptive scheme is based upon assigning a hierarchial system of labels to the points which make up the picture. At each level the labeled outputs from lower levels serve as input to the current level of labeling. At the lowest level (executed by the scanning unit) each point is assigned one of the labels, black or white. In the next level (executed by the PAU) all (say) black points are locally connected together in road-like elements and assigned an orientation: north-south, east-west, right-diagonal, left-diagonal.

Using the road segments so formed, higher order "phrases" are formed--the sublists generated from each window. But at this level of synthesis it is increasingly obvious that the rules of grammar (syntax) for optimally joining elements together in the list structure are highly class dependent: the description of bubble-chamber negatives and the description of handwriting, though both composed of road-like elements, clearly part at this level if not earlier.

---

* From the Greek: ταξισ: rank, arrangement (originally, pattern) and κρινω: to judge.

It would not serve our purpose to elaborate further here the Syntactic Model of visual recognition. This model, however, was originally developed to answer a very specific problem: the identification of patterns that occur in bubble-chamber negatives. Toward this end a prototype high-resolution oscilloscopic scanner was constructed and to firm the design a simulator, of the Pattern Articulation Unit for the IBM 7094 has been written and provided with a fairly complete set of programming macro-instructions.[3,4] Using these tools a variety of specific labeling and list processing algorithms have been developed.[8,10,11] A comprehensive program for bubble-chamber automatic scanning is currently being written.

3. REALIZATION OF THE PATTERN ARTICULATION UNIT

3.1 The Iterative Array

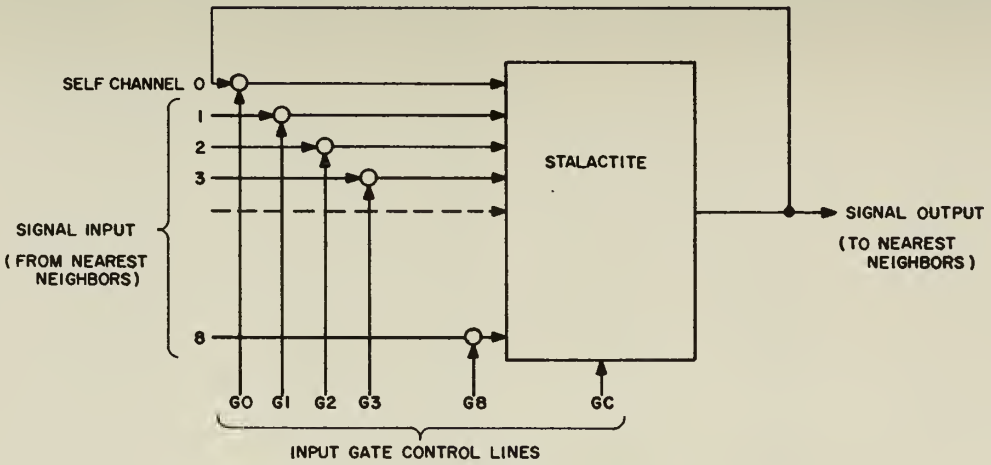
The Pattern Articulation Unit (PAU) is a parallel processing unit for the automatic scanning of bit-encoded visual images. The iterative array of the PAU consists of 1024 identical processing modules (Fig. 7) called stalactites, placed at sites of a two-dimensional Bravias net (32 x 32). Accordingly the internal word size in this processor of the computer is 32 x 32 = 1024 bits. The logical design of the all-digital processor has been optimized for the idealization of the input image to a line drawing. Nodes representing end points, bends, points of intersection, etc., of this idealized line drawing are then labeled by appropriate programming. Finally, the abstract graph describing the interconnection of labeled nodes is extracted as a list structure, which comprises the normal output of a processing unit.

Initially the digitized input image is transferred line-by-line to the stalactite array. It is observed from empirical simulation studies that the input image (window) can be idealized to a line drawing by a sequence of simple boolean functions of the output of nearest neighbors. That is, the new value ("0" or "1") assigned to a site is a specified boolean function of the output of neighboring sites. In particular the array can function as a two-dimensional shift register. Neither the selection of relative neighbors involved nor the boolean dependence is allowed to change site-to-site. Accordingly, common control signals are sent to all modules. The number of control lines is independent of the array size (i.e., number of stalactites).

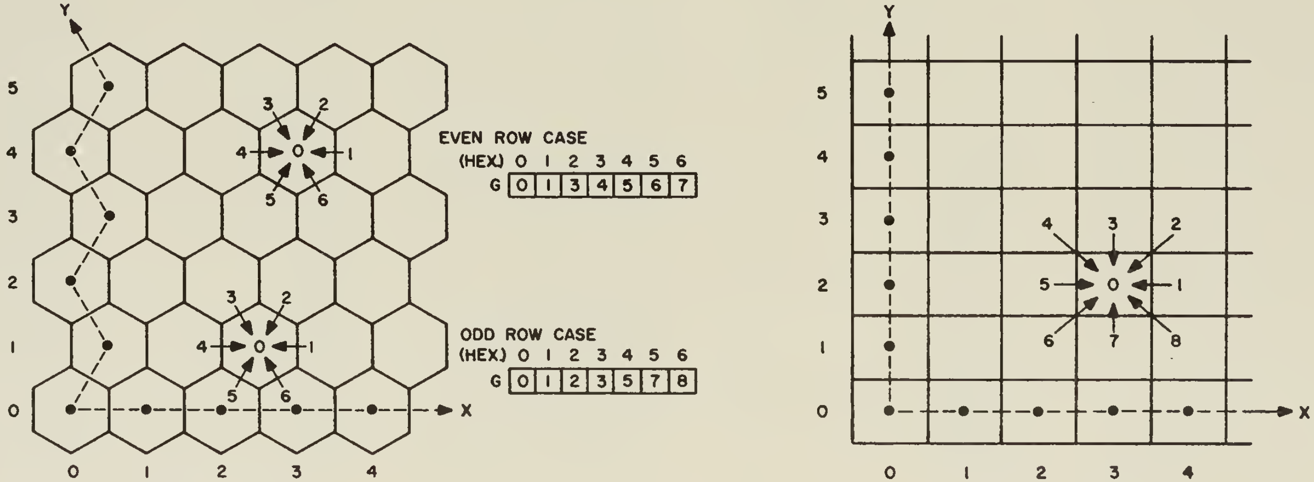
3.1.1 Thinning

Parallel preprocessing of the window image, the noise-cleaning operations mentioned above, can be largely accomplished by means of homogeneous logical transformations, i.e., by means of specific boolean operations performed in parallel on the immediate neighbors of each digitized point of the picture.

Elementary homogeneous logical transformations can be subdivided into symmetric and asymmetric boolean functions of the content of neighboring cells. Symmetric functions, including the large class of matching operations, implicitly reduce to a counting operation, and are treated in the next paragraph. Asymmetric boolean functions occur more commonly in the context of thinning operations.



SIGNAL & INPUT GATE CONTROL LINES OF STALACTITE



RHOMBIC ARRAY

RECTANGULAR ARRAY

CONNECTIVITIES OF THE ITERATIVE ARRAY

FIGURE 7

Intuitively the idea behind digital image thinning is to "shrink" an original digitized image to a line drawing--while preserving all connectivity of the original. The utility of the thinning process is readily apparent. Thinning extends to the local neighborhood of a point certain global aspects of the picture. For example, imagine a process by which all black track-like elements are thinned to "unit" thickness, massive black areas are reduced to mere peripheries, while intersection junctions of the original are preserved. For a rectangular (or rhombic) raster we can then classify all black cells of this "thinned" image into four categories--merely by counting the number of black points in the eight (six) cells immediately surrounding the cell in question:

Class of Point	Number of Black Points in Immediate Surround
isolated point	0
end point	1
interior point (of a line segment)	2
junction point	3 or more

This mode of classification, applicable only for thinned images, is an immense simplification of the equivalent procedure for unthinned images. Positions noted by eye as end, interior, or junction point on the original (unthinned) digitized image are selected, in general, on the basis of elaborate decision criteria--often involving a knowledge of neighboring points over an extensive area.

Simulation studies to date show that line width normalization (thinning) can be efficiently realized by simple AND, OR, NOT boolean functions of the output of nearest-neighbor stalactites.

3.1.2 Bubble Logic

Certain noise-cleaning operations, in particular gap filling, implicitly involved an estimation of size--a counting operation.

The neighboring bits in question are transferred to eight bits of the internal storage register of the stalactite. These cells of the stalactite are made "self-ordering"--a fixed number of identical operations will order the contents of the register so that all "zeros" rise to the top, all "ones"

precipitate to the bottom of the vertically-aligned stalactite (Fig. 8). Once in this canonical form, threshold logic is achieved by sampling (in parallel) contents of all internal flipflops of given depth.

For a physical description of the bubbling process consider a vertically-oriented register containing mixed "1's" and "0's." Examine all adjacent pairs of digits and everywhere replace each $\binom{1}{0}$ pair by a $\binom{0}{1}$ pair. All other digits remain unchanged. Clearly this operation neither creates nor destroys "1's" but, if iterated, "1's" ultimately move to the bottom-most cells. For an r-bit register a maximum of r - 1 iteratives of the bubble operation are required to order completely the contents of the register.

Figure 8 shows an example of bubbling in which four operations completely order the register. Subsequent operations do not change the contents of the register so it is not necessary to sense the completion of ordering. The register is shown oriented vertically to strengthen the analogy that "0's" are bubbles moving to the top, hence the term "bubbling."

To describe the logical function involved let A, B, C be any three consecutive positions of the register (counting downward). The basic bubbling operation (B1D) is then given by

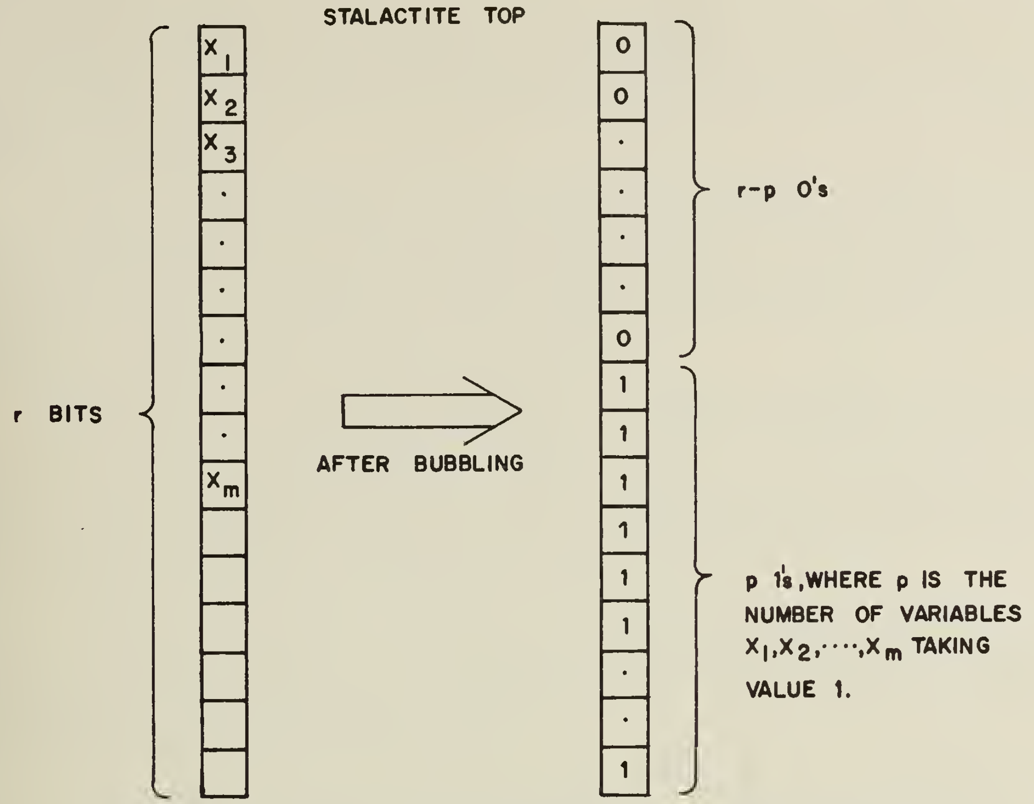
$$\text{Bubble 1's Downward (B1D)} \begin{cases} \text{top} & B' = BC \\ \text{all interior} & B' = A\overline{B} \vee BC \\ \text{bottom} & B' = A \vee B \end{cases}$$

where B is any register position, A and C are adjacent positions (A above, C below) and B' is the new contents of the register B.

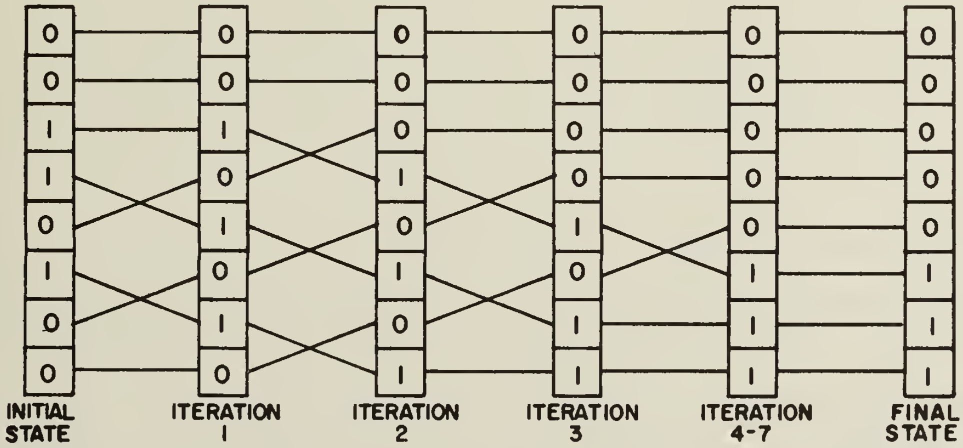
In addition bit stacking (and unstacking) facilities are built into the stalactite logic. These prove to be special cases of the bubble logic.

3.1.3 Structure of the Stalactite

The iterative array of stalactites can be connected in either rectangular or rhombic symmetry at programmer's option (Fig. 7). Each stalactite can be visualized as having a sole transmitting channel (the signal output line). In turn the stalactite can receive information from any (or all)



CANONICAL FORM FOR SYMMETRIC BOOLEAN FUNCTIONS OF m VARIABLES($m \leq r$) FIGURE 8A



SUCCESSIVE STEPS IN THE BUBBLING PROCESS
FIGURE 8B

of its immediate neighbors: eight for a rectangular array, six for a rhombic array (see Fig. 7). In addition the stalactite can sense its own output (input channel 0 in Fig. 7).

The internal symmetry of each stalactite module reflects all translational, rotational, and reflectional symmetries of an (infinite) planar array, and employs a minimum number of active elements consistent with efficient realization of processing macro-instructions.

The generic stalactite contains ten one-bit registers of internal storage (Fig. 9). These flipflops, labeled M, 0, ..., 8 serve as (1) the buffer register of the transfer memory (M); (2) an auxiliary nonbubbling register (0); and (3) eight consecutive bubbling flipflops (1, 2, ..., 8).

If we ignore intra-stalactite processing (bubbling, etc.), the simple structure of the stalactite for communicating with its neighbors can be understood from the following constraints:

1) the new value of every internal flipflop is a function of only the normal nine inputs (eight neighbor, one self) and of control lines common to every stalactite of the array.

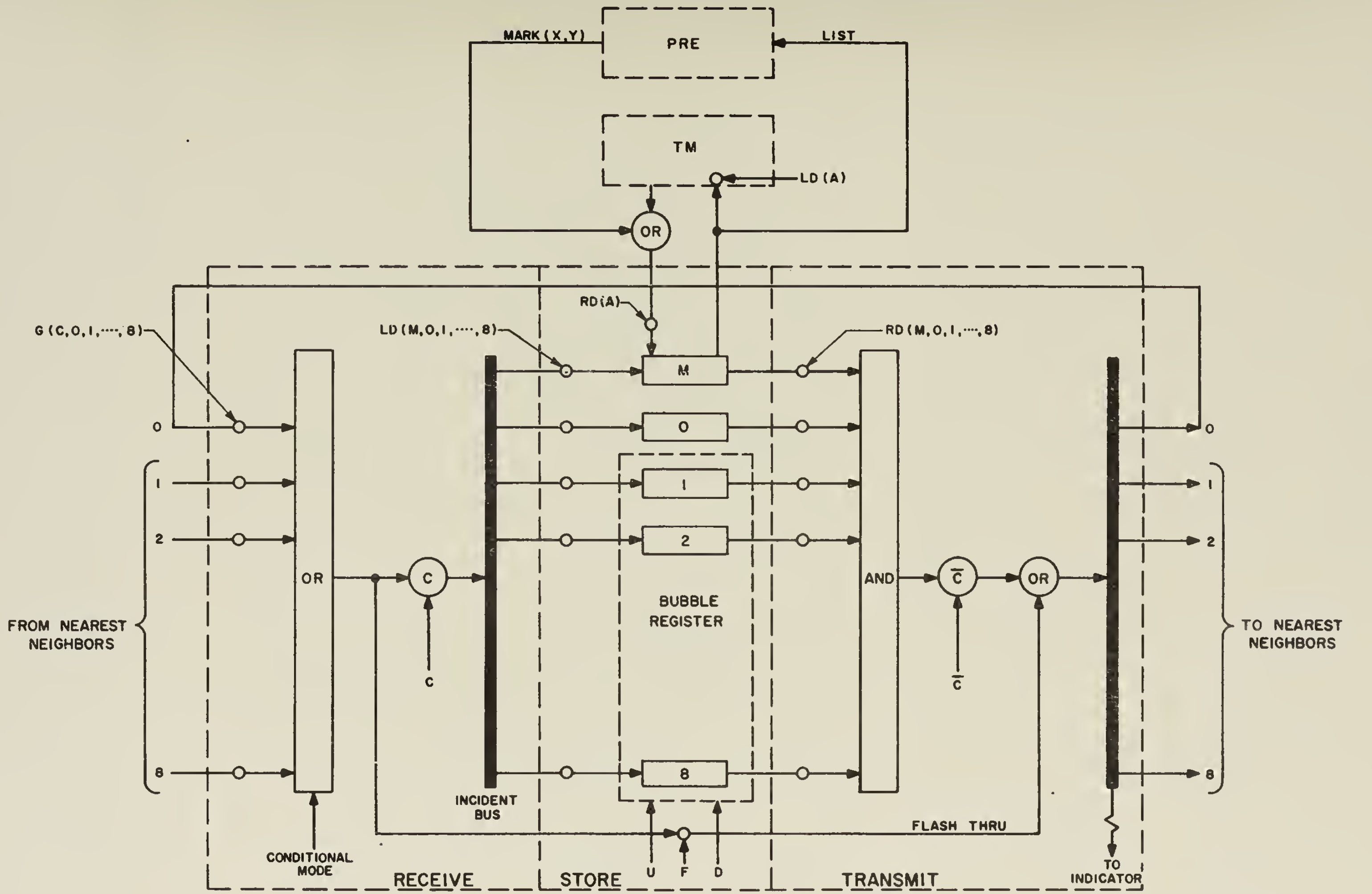
2) the new input signal is a function only of the present contents of the internal flipflops and of control lines common to every stalactite of the array.

In addition to avoid race conditions, we will insist

3) in any one instruction an internal flipflop is not both read from and written into. (Of course in any one instruction flipflops may be inert and ignorable: neither read from nor written into.)

Constructions of the above type can be called unconditional transfer orders--unconditional, because the execution of the instruction does not depend on the internal state of the stalactite.

Let us partition the generic stalactite into three blocks labeled TRANSMIT, RECEIVE and STORE as shown in Fig. 9. Blocks TRANSMIT and RECEIVE, as implied by the constraints (1) and (3) above, are pure combinatorial nets (i.e., without internal storage).



SCHEMATIC OF GENERIC STALACTITE

FIGURE 9

A given internal flipflop can in any one machine instruction be written into, or read from, but not both. That is, the RD control signals and the LD control signals must be partitioned into two disjoint sets. Alternatively one could partition the set of internal flipflops, marking some for "intermediate storage" as in the traditional double-ranked shift register. We will find it faster (one machine cycle rather than two) to have the control assign (in common for the whole array) internal flipflop numbers to intermediate results, and permute these assignments from one instruction to the next.

The ten registers of internal storage, the block STORE, are not differentiated, and therefore fed from a common bus. The TRANSMIT combinatorial net must be a symmetrical boolean function of the output of the ten internal registers, if these are to be undifferentiated. The AND function of outgated flipflops, with or without subsequent complementation has been chosen for the TRANSMIT net.

That the net RECEIVE must be a symmetric net of its nine inputs is less obvious. We have proved a theorm that asserts "a generic stalactite, capable of being embedded in an (ideally infinite) array of either rectangular or (at programmer's option) rhombic array, and exhibiting all translational, rotational, and reflectional symmetries of the embedding space, must be a symmetric function of its eight input terminals (from nearest neighbors)." The 0 or self-input terminal is shown to be distinguished by this proof, and could be treated separately. Notice that inputs from diagonally-connected neighbors are in no way distinguished, even though for the rectangular array alone there exists no rotation or reflection interchanging diagonal and horizontal/vertical axis inputs. This symmetry is in force only because the stalactite array can be viewed (at the option of the programmer) with either rectangular or hexagonal packing. The OR function of all ingated signal lines, with or without subsequent complementation, has been selected for the RECEIVE net.

3.2 The Transfer Memory[15]

The central core of the pattern articulation unit (PAU) has been introduced above as an iterative array of processing modules with word size 32 x 32. These modules require individual access to memory. This memory can be partitioned into a part intrinsic to the stalactite, i.e., those flipflops implicit in the realization of threshold logic, etc.; and into a remaining part

extrinsic to the iterative array proper. Collectively this latter memory will be referred to as the transfer memory (TM) and is logically inert.

3.2.1 Use of Auxiliary Memory

Demands for intermediate storage are seen to arise naturally--either (1) from the nature of the input visual image, or (2) from the necessity to retain intermediate partial results. Examples of the former situation arise when successive input images are continuous in (i) time sequence (neighboring motion picture frames, television scans, etc.), (ii) tone (successive slices of gray scale), or (iii) color (the digitization implicity in four-color printing). Here congruent storage is essential if the processing routine is to exploit similarities of image shape from frame to frame.

The necessity to store intermediate results of the iterative array stems directly from the means of executing honogeneous logical transformations in the processor. The complexity of the iterative array can obviously be greatly simplified if various digital filtering operations can be performed not in parallel, but serially, and intermediate results (32 x 32 = 1024 bit words) stored. For example, local track setment orientation can be designated as being predominantly horizontal, vertical, left-diagonal or right-diagonal in four serial tests, whereas simultaneous identification requires approximately four times as much hardware.*

To this end the transfer memory supplies 48 planes of 1024 bits each of random access memory. In this mode of operation one bit from each of the 1024 processing modules (stalactites) is transmitted in parallel to (from) the TM in any one memory cycle.

---

* The intrinsic speed advantage of contemporary digital circuitry over neutral response rates strongly suggests that a different schema--time-multiplexed use of one processing array with facilities (TM) for storage of intermediate results--be used in the man-made recognition machines, in contrast to the extreme filtering parallelism found, for example in the frog's eye. [H. R. Maturana, J. Y. Lettvin, W. S. McCulloch, and W. H. Pitts, "Anatomy and Physiology of Vision in the Frog," Jour. General Physiology, 43, 129, (1960)]

3.2.2 Extraction of Information at a Labeled Point

Normally a processing routine can be case into a form where relatively few stalactites are labeled, i.e., singled out for detailed examination. All tracking operations culminate in this form--labeling of terminal nodes, points of intersection, etc. The essential point is that ancillary information--most naturally associated with each labeled node--can be extremely useful. For example in bubble-chamber data processing local orientation can help distinguish track segments of an electron spiral from segments of an intersecting beam track.

Such information, perhaps aimed at improved tracking by assisting selection of the next point of readout, is extracted from the transfer memory by a second mode of reference. Rather than a global view of one bit from each of the 1024 stalactites, parallel output of the contents of one stalactite is sampled. In this latter node the transfer memory is viewed as a random access memory of 1024 words, one word (48 bits) associated with each cell of the processing array.

3.3 List Compilation

3.3.1 List and Mark Instructions

The problem of extracting an abstract graph description of a line drawing embedded in a plane of the stalactite array has been introduced in Section 2.2. Once each node of the line drawing has been labeled in parallel by local boolean processing, the list compilation facilities required to read out (and thus generate the abstract graph) can be reduced to two principal operations:

1) List Instruction. All points (i.e., black cells of a sparsely populated plane) are sequentially identified by ten-bit coordinates Z = (X,Y) with five bits X, five bits Y. Sequential readout implies a linear ordering of all 1024 cells of the array, and that an algorithm for finding the first black point is known.

2) Mark Z Instructions. A single cell specified by address Z is marked (i.e., set = "1"). Marking, as will be shown below, can also be used for erasing points of the sparsely

populated plane as found. Iteration of the search algorithm to find the "first" black point, alternated with subsequent erasure of each point as found, eventually lists all points of the plane.

3.3.2 Use of the M Register

From engineering considerations* it is convenient to specialize the mark and list instructions to the M plane of the stalactite array. The two coordinate instructions then take the form:

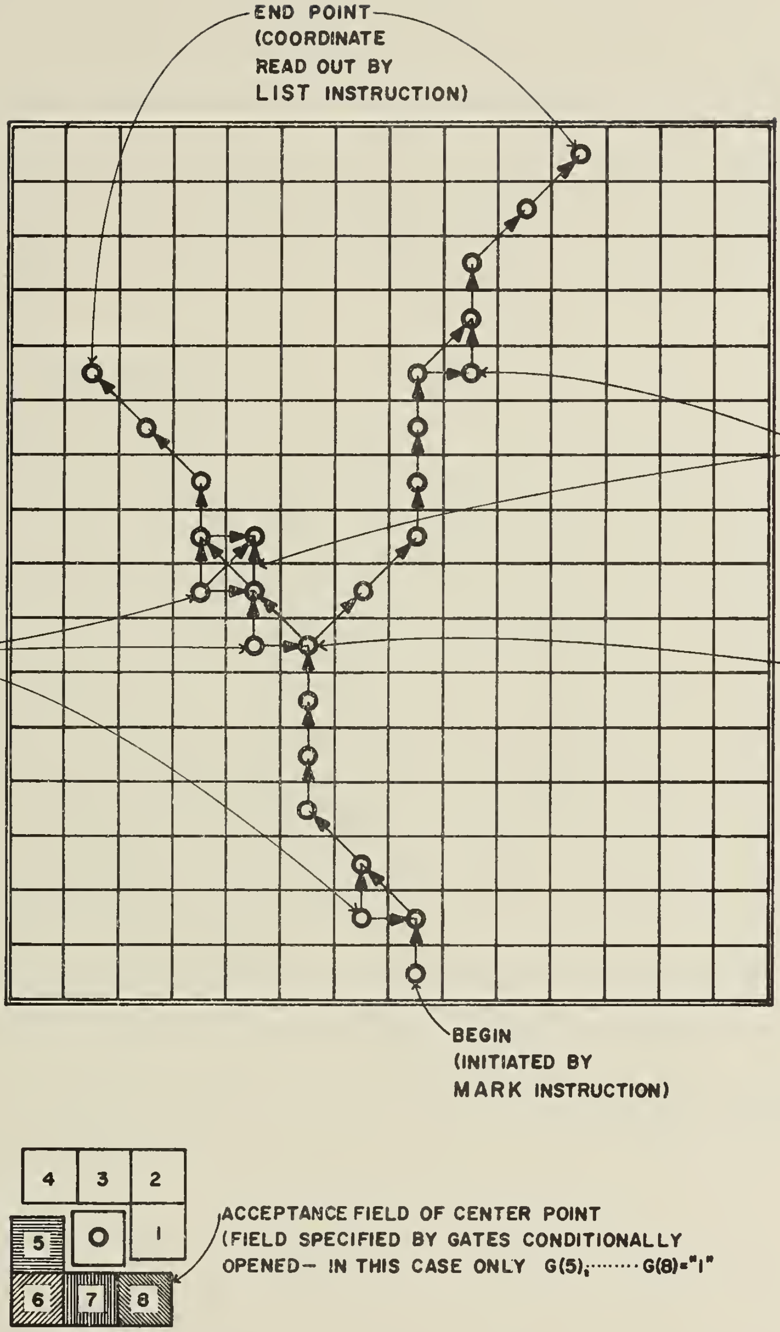
MARK (Z):	sets to "1" the M flipflop of the stalactite at coordinate Z = (X,Y).
LIST:	serially compiles a list of all coordinates Z of stalactites having "0" in the M flipflop.** Coordinate Z of the next point found is transferred to the LIST register automatically upon storage of the previous contents of the register.

In use, a sparsely populated plane is complemented, loaded into the stalactite M register, and a first black point identified by the LIST instruction. Each coordinate of this list, once compiled, is used to MARK its associated M flipflop (and thus erase the point). Black points connected to this point (found by flash-through) are in turn listed. This technique (for isolated track segments each LIST instruction brings down only one entry) builds a paired list of nodes, i.e., the branches of the associated abstract graph (Fig. 10).

---

* Coaxial cable connects the transfer memory (TM) and the M plane. It is convenient to make the MARK and LIST instructions adjuncts of the TM gating arrangement and to capitalize upon the close jaxtaposition of M-plane signal lines at the TM.

** As we normally complement a plane before transferring it to the M register and initiating the LIST instruction, we will refer to a "0" in this M plane as a "black point."



END POINT (COORDINATE READ OUT BY LIST INSTRUCTION)

NOTE DUAL PATHS PERMISSIBLE

INERT POINTS (NO ENTRANCE UNLESS SPECIFIED BY MARK INSTRUCTION)

PATH CAN BE TERMINATED HERE BY REMOVAL OF THE JUNCTION POINT

BEGIN (INITIATED BY MARK INSTRUCTION)

ACCEPTANCE FIELD OF CENTER POINT (FIELD SPECIFIED BY GATES CONDITIONALLY OPENED— IN THIS CASE ONLY G(5),........G(8)="1"

PATH BUILDING WITH MULTI-SELECTION SWITCH ARRANGEMENT

FIGURE 10

3.3.3 Search Algorithm Ordering

The list instructions as mentioned above imply a linear ordering of all 1024 cells of the two-dimensional array. A point in this array is specified by coordinate (X,Y) with five bits of X, five bits of Y. Rather than the customary integer representation it is convenient to visualize $(X + iY)$ as a point within the unit square: $0 \leq X < 1$, $0 \leq Y < 1$ with X,Y proper binary fractions. All "M" cells with an output signal of "0" are lexicographical ordered first on Y, then X. The cell of lowest order is then LISTed first (Fig. 10).

3.3.4 An Associate Memory

Coordinate readout, in conjunction with the logical processing facilities of the stalactite array, combined to transform the transfer memory into a versatile associative memory. Consider the transfer memory initially filled with ≤ 1024 words, each of 48 bits. Let any specified subset of bit positions (0, 1, ..., 47) be designated a code field. The associative property in simplest form asserts that the addresses of all words (if any) having as contents of their code field an (arbitrary) requested bit pattern are directly identified. In the PAU design parallel associative search is done by transferring the bit plane of the code field in turn to the stalactite array with appropriate complementation such that the ideal pattern, so manipulated, is identically zero. After stacking (or bubbling), the M bit of all blank stalactites is marked and the corresponding addresses compiled. The logical flexibility of the stalactite array permits obvious generalizations of this technique: in particular, the addresses of those words whose code field bits satisfies some simple boolean function can be selected by appropriate intra-stalactite programming.

3.4 PAU Order Code

For purposes of control the instruction set of the PAU can be partitioned into seven basic types of instructions:

1) stalactite transfer orders (conditional/unconditional, with/without simultaneous border transfer)

2) stalactite bubble orders

3) transfer memory orders

4) flash-through

5) pyramidal readout (i.e., list, mark instructions)

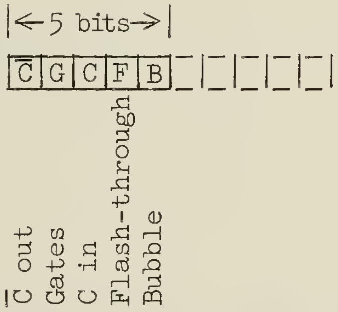
6) border input/output

7) special instructions

In the interest of brevity we will treat in detail here only the stalactite transfer and bubble orders. Instructions requiring communication between the stalactite array and the transfer memory will be momentarily ignored in this section. We emphasize that the instructions described below are machine micro-instructions. Programming macro-instructions, on the other hand, are discussed in references 3 and 6. Sample code is illustrated in Fig. 11.

For pedagogical reasons it is convenient to introduce two redundant bits and subdivide the resulting 40-bit stalactite μ-instruction into four control groups of ten bits each. The first of these is called the flow control as the first five bits of the group specify the information flow through the stalactite as a whole.

Flow group (first five bits):



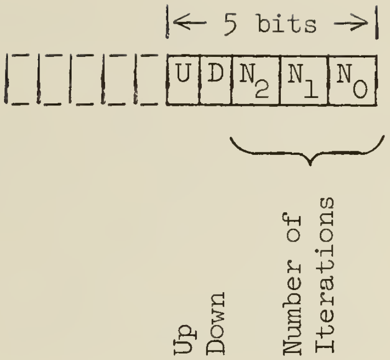
Here $\bar{C}$ and C specify whether the outgoing or incoming signal respectively, or both, are to be complemented (see Fig. 9). Bit G, when "1" and only then, indicates that the gate input control group must be examined as at least one input gate is opened (gate control $\neq 0$). Bit F, whe "1," indicates the flash-through bypass is to be called into play.

The fifth, or B bit, when "1," evokes a bubble logical operation, the selection of which is given by the second five bits of the flow group.

```
               FORNANGO, J. P.                                           31022

•
•  INITIAL PRE-PROCESSING ROUTINE FOR BUBBLE CHAMBER PICTURES
•
       SETMOD  72
       LETTER  ( X)
       BCOLOP  MO,,CLEAR,,,,,BAR        COMPLETELY FILLED CONTEXT PLANE
START  READ    M1
       PRINT   M1,,72,72
       CCCMMT  (-ORIGINAL PICTURE)

•••••••     FILLING ROUTINE     •••••••

       INDEX   SET,1,0
FILL   BCOFUN  M2,,MO,,M1,,(37$)        FILL SINGLE HOLES IN N. DIR.
       BCOLOP  M1,,OR,M1,,M2
       TMARK   DLSUR,,M2,,MO,,M1,,GE,7  FILL HOLES WITH 7 OR 8 NEIGHBORS
       BCOLOP  M1,,OR,M2,,M1
       BCOFUN  M2,,M1,,M1,,(2/5/6/7$)   CORNER FILL
       BCOFUN  M2,,MO,,M2,,(5+7$)
       BCOFUN  M3,,M1,,M1,,(4/7/8/1$)
       BCOFUN  M3,,MO,,M3,,(7+1$)
       BCOFUN  M4,,M1,,M1,,(6/1/2/3$)
       BCOFUN  M4,,MO,,M4,,(1+3$)
       BCOFUN  M5,,M1,,M1,,(8/3/4/5$)
       BCOFUN  M5,,MO,,M5,,(3+5$)
       BCOLOP  M1,,OR,M1,,M5
       BCOLOP  M1,,OR,M1,,M4
       BCOLOP  M1,,OR,M1,,M3
       BCOLOP  M1,,OR,M1,,M2
       INDEX   INCR,1,1
       JUMP    LESS,FILL,1,2
       BCOFUN  M2,,M1,,M1,,(5/1$)       EAST FILL
       BCOFUN  M3,,M1,,M1,,(1/5$)
       BCOLOP  M2,,OR,M2,,M3
       BCOFUN  M2,,MO,,M2,,(15$)
       BCOLOP  M1,,OR,M1,,M2
       INDEX   SET,1,0

•••••••     THINNING ROUTINE     •••••••

       BOOLOP  M7,,EQUAL,M1
THIN   TMARK   DLSUR,,M2,,M1,,M1,,LE,3  SAVE POINTS WITH 3 OR LESS NEIGH
       BCOFUN  M3,,M1,,M1,,(37$)
       BCOFUN  M4,,M1,,M1,,(15$)        REMOVE P IF IT IS A BOUNDARY PT
       BCOFUN  M5,,M1,,M1,,(3/7+7/3$)         NEXT TO 2 INTERNAL PTS.
       BCOFUN  M6,,M1,,M1,,(1/5+5/1$)
       BCOFUN  M3,,M3,,M3,,(1+5$)
       BCOFUN  M4,,M4,,M4,,(3+7$)
       BCOFUN  M5,,M5,,M3,,(3+7$)
       BCOFUN  M6,,M6,,M4,,(1+5$)
       BCOLOP  M3,,OR,M5,,M6
       BCOLOP  M4,,AND,M1,,M3,,BAR
       TMARK   DLSUR,,M5,,M2,,M3,,EQ,2  PUT BACK SAVED PTS WHICH DID NOT
       BCOLOP  M6,,AND,M2,,M5,,BAR            HAVE 2 NEIGHBORS REMOVED.
       BCOLOP  M2,,OR,M6,,M4
       BCOFUN  M1,,M7,,M2,,(26+48$)     PUT BACK REMOVED PTS WHICH WOULD
       BCOLOP  M1,,OR,M1,,M2                  BE DIAG. INTERNAL IF KEPT.
       INDEX   INCR,1,1
       JUMP    LESS,THIN,1,3
```

Figure 11. Sample PAU Simulator Program[13]
(Using Macro-Instruction Set)

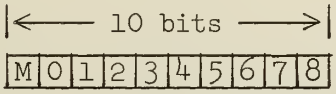
Flow group (second five bits):



Bits U and D (positions 6 and 7) define the type of bubble logic. Here the number of iterations to be applied (0, 1, 2, ..., 7) is given by the binary number $N_2N_1N_0$. When B = 1, the three bits of iteration number N are transferred to a control counter to control iterations of bubbling.

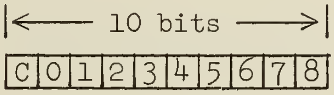
Having set up the flow it is appropriate to interrogate which flipflop outputs are to be sent as a confluence (AND/$\overline{\text{AND}}$) out over the output channel. The appropriate flipflops are specified by the <u>FF Read Control group</u>:

FF Read Control group:



Having thus established the output signal of every stalactite we sum (OR) together the requisite input signals to the stalactite, as specified by the <u>Input Gate Control group</u>:

Input Gate Control group:



Here the significance of the C bit needs re-emphasis: C (= "1") signifies that conditional input gating is to be used, i.e., each gate 0, 1, ..., 8 marked with a "1" in the input gate control group is, in fact, open <u>only</u> if the correspondingly numbered stalactite flipflop contains a "1" at the beginning of this instruction.

It will be remembered that the iterative array can operate in two connectivities: rectangular or rhombic. For the rectangular case all ten bits of the input gate control group are used; for the rhombic array, the last two

bits are ignored and the input gates, as associated internal flipflops, renumbered to conform to the conventions.

Finally, the output signal of the input combinatorial net (called the <u>incident bit</u>) is, after possible complementation, ready to be stored, or interrogated as the conditional control bit for a bubble-logic operation (if specified by B = "1"). If the former situation obtains, then and only then, is the fourth and final control group examined--the <u>FF Load Control Group</u>:

FF Load Control group: |← 10 bits →|

M	0	1	2	3	4	5	6	7	8

Here each of the ten bits corresponds respectively with the ten internal flipflops of the generic stalactite.

In summary the intra-stalactite micro-instructions are specified by four control groups of ten bits each, placed in the 40-bit stalactite instruction register (SIR):

Stalactite Instruction Register:

I	II	III	IV
10	20	30	40
Flow	FF Read	Input Gate	FF Load

The interpretation of the individual control groups has been described above.

3.5 <u>Fabrication of the Pattern Articulation Unit (PAU)</u>

3.5.1 <u>Layout</u>

From a packaging point of view the PAU consists of six subdivisions listed below:

1) iterative array
2) PAU control
3) transfer memory and pyramidal readout encoder
4) SCR power supplies

5) final voltage regulators

6) air-conditioning unit

All units are housed in the computer main frame with the exception of the SCR power supplies (basement), relay wall boxes (wall-mounted in computer room), and air-conditioning unit (roof-mounted directly above main frame). The distribution of the remaining units as housed in the main frame is shown in Fig. 12. Each of these units is discussed momentarily below.

3.5.2 The Iterative Array

There are six basic circuit boards of the iterative array. Each circuit board is packaged on a 3.984"-wide x 4.750"-long photoceram board. A total of 1360 boards are required. Of these 1024 are used to house the stalactites of the iterative array, one per board.

The distribution of circuit boards of the iterative array is housed in the four front transistor bays of the computer main frame is shown in Fig. 13. Control drivers are placed largely with middle of the transistor bays to keep wiring length to a minimum. Drivers to even and odd rows of the array are separated, consistent with the potentiality of using either rectangular or rhombic connectivity.

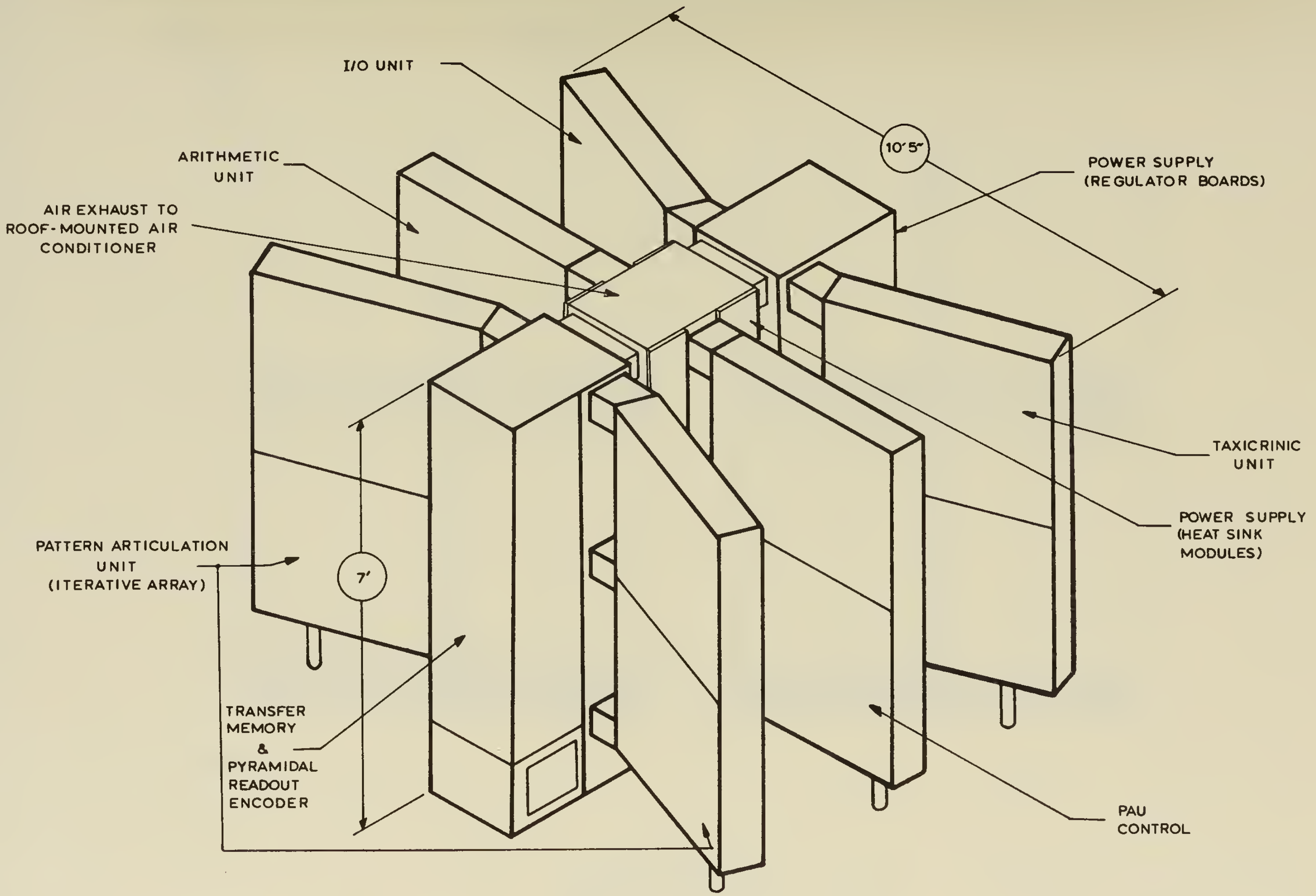
Every stalactite board has two coaxial connections (respectively to/from the TM, and hence PRE). This integrates to a massive arterial network of miniature coaxial cable which is fanned into the TM (front central rack). Coaxial connections to (from) the border of the 40 x 40 array, though extensive, are nominal in comparison.

3.5.3 The Stalactite

Because the stalactite plays a role of central importance in the array, its design configuration is shown in Fig. 14.*

---

* This circuit design of Professor K. C. Smith will be separately reported in more detail by him at a later date (see reference 18).



ILLINOIS PATTERN RECOGNITION COMPUTER (ILLIAC III)

figure 12.

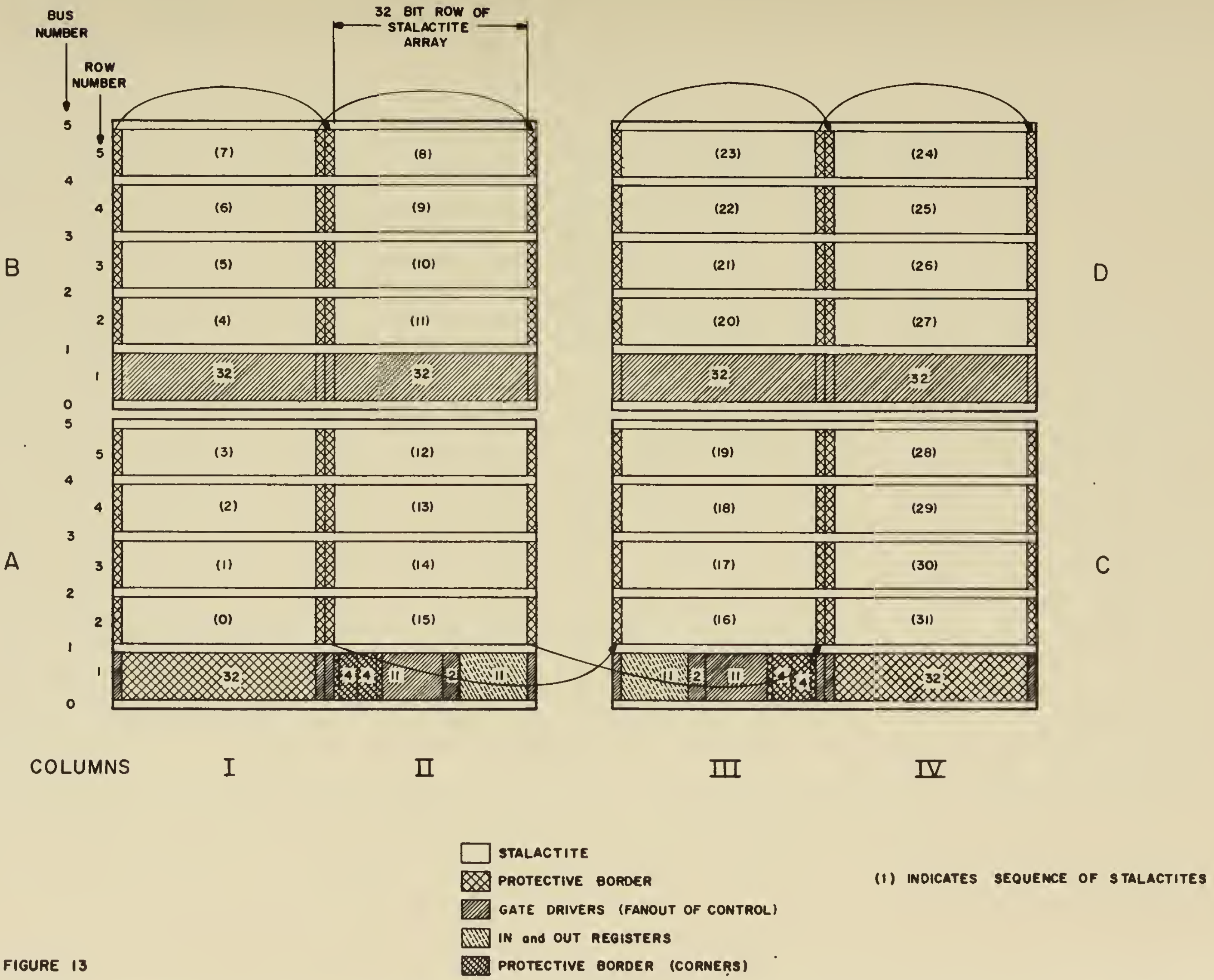


FIGURE 13

LAYOUT OF PAU ITERATIVE ARRAY

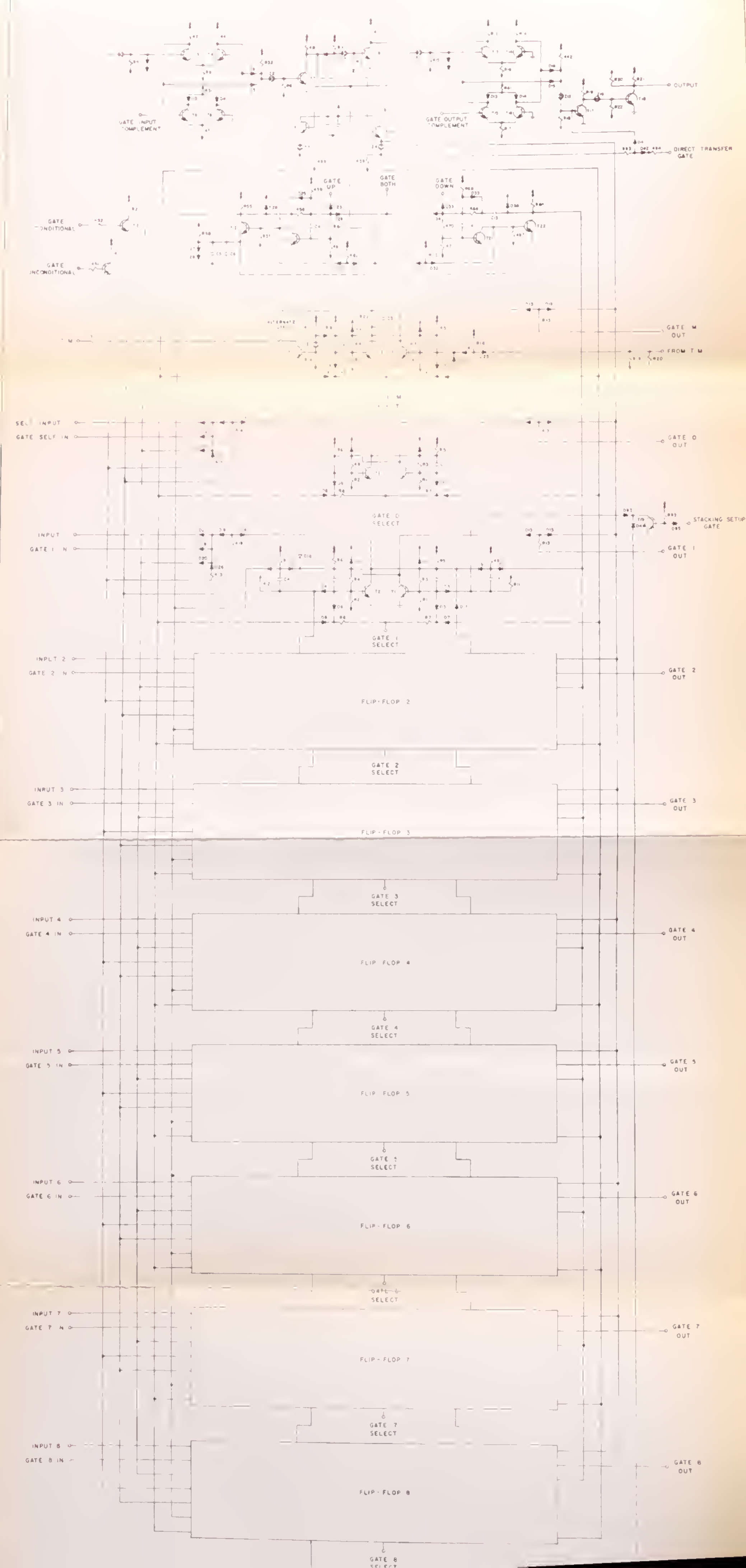
ol,

nal
n
ors

ete

as
e

,


GATE INPUT COMPLEMENT
GATE OUTPUT COMPLEMENT
OUTPUT
DIRECT TRANSFER GATE
GATE UP
GATE BOTH
GATE DOWN
GATE CONDITIONAL
GATE UNCONDITIONAL
T M
GATE M OUT
FROM T M
SELF INPUT
GATE SELF IN
GATE 0 OUT
GATE 0 SELECT
STACKING SETUP GATE
INPUT
GATE 1 IN
GATE 1 OUT
GATE 1 SELECT
INPUT 2
GATE 2 IN
FLIP-FLOP 2
GATE 2 OUT
GATE 2 SELECT
INPUT 3
GATE 3 IN
FLIP-FLOP 3
GATE 3 OUT
GATE 3 SELECT
INPUT 4
GATE 4 IN
FLIP FLOP 4
GATE 4 OUT
GATE 4 SELECT
INPUT 5
GATE 5 IN
FLIP FLOP 5
GATE 5 OUT
GATE 5 SELECT
INPUT 6
GATE 6 IN
FLIP-FLOP 6
GATE 6 OUT
GATE 6 SELECT
INPUT 7
GATE 7 IN
FLIP-FLOP 7
GATE 7 OUT
GATE 7 SELECT
INPUT 8
GATE 8 IN
FLIP-FLOP 8
GATE 8 OUT
GATE 8 SELECT

Certain features of the design warrant emphasis. First it should be observed that virtually no decoding of control signals is done within the stalactite. That is, each control line is directly identified with a bit in the general micro-instruction.

Secondly, almost half the circuitry of a module is expended on control, or routing of the signal, reflecting the wide range of admissible orders that a module can execute.

That portion of the stalactite of high symmetry, i.e., the ten internal flipflops with associated gates and bubble logic, are realized in ten thin-film microcircuits, each 0.42 x 2 inches (Fig. 15).* Conventional silicon transistors and microdiodes are used. Parts of low symmetry, i.e., the control portions of the stalactite, are packaged on printed-circuit chips using conventional discrete components.

3.5.4 Transfer Memory and Pyramidal Readout Encoder

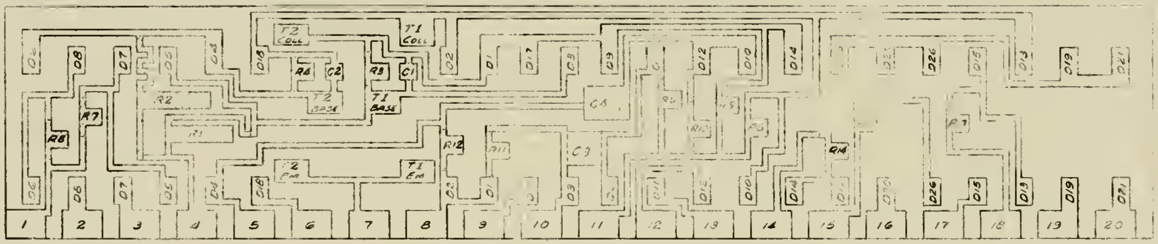
The transfer memory and the pyramidal readout encoder function together as an associative memory. For this reason they are packaged together in the entire front center rack and share the 1024 coaxial cables entering from and the 1024 coaxial cables exiting to the 1024-bit M register of the stalactite array.

Packaging the circuitry of the transfer memory in close superposition with the core matrix stack requires considerable ingenuity. A drawer arrangement holding the transfer memory circuitry, pyramidal readout encoder circuitry, the matrix stack, as well as the interconnecting coaxial cable, is planned.

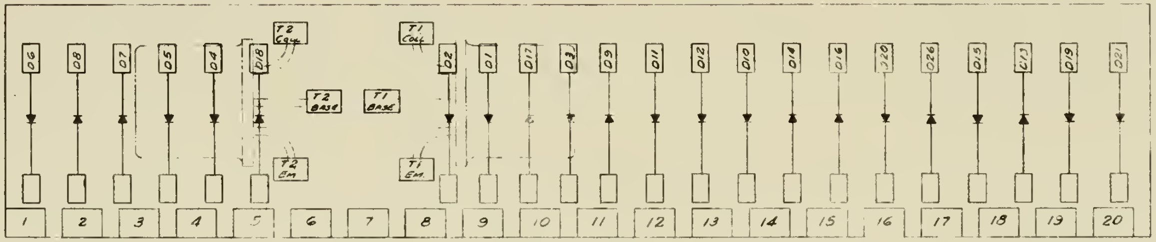
---

* Dr. Hans Bilger materially assisted in these layouts.

1 Gate Select
2 Signal Driver-Left
3 Signal Driver Right
4 −6 Volts
5 Flip Flop Below Left
6 Flip Flop Above Left
7 Ground
8 + [illegible]
9 Flip Flop Below Right
10 Flip Flop Above Right
11 12 [illegible]
12 [illegible] Left
13 Use [illegible] Right
14 [illegible]
15 Con[illegible] Left
16 [illegible] Complement
17 To Output [illegible] Complement
18 Gate Out
19 Gate In
20 Input



Conductor, Passive Component Layout



Location Of Diode & Transistor Attachment

Figure 15

BHM PWR. 8/19/63 PLANAR LAYOUT FLIP FLOP "I" STALACTITE SHEET 1 OF 3 D-3097

4. SUMMARY

This report has described the system design of an all-digital computer for visual recognition. One processor, the Pattern Articulation Unit (PAU), has been singled out for detailed discussion. Other units, in particular the Arithmetic Unit and the Taxicrinic Unit, are treated in reports listed in the attached bibliography.

The PAU has been shown to be a processor of fundamentally new design--its logical organization has no analogue in the central processing unit of existing computers. The PAU is the first modular parallel processor (1) which because of its digital organization is capable of more reliable visual identification than part analogue/part digital preprocessors of much less generality and potential virtuosity; (2) which is faster than any presently suggested alternative realizable today at comparable cost; (3) and which can serve as a prototype to a new generation of parallel computers that will capitalize upon thin film and integrated semiconductor circuitry of the immediate future.

SELECTIVE BIBLIOGRAPHY OF PAPERS
ON THE
ILLINOIS PATTERN RECOGNITION COMPUTER (ILLIAC III)

Summary Articles

☐ 1. B. H. McCormick and R. Narasimhan, Design of a Pattern Recognition Digital Computer with Application to the Automatic Scanning of Bubble-Chamber Negatives, Nuclear Instruments and Methods 20, (1963), 401-406.

☐ 2. B. H. McCormick, The Illinois Pattern Recognition Computer (ILLIAC III), Digital Computer Laboratory Report No. 148, August 20, 1963

Programming (PAU Simulator)

☐ 3. James H. Stein, User's Manual for PAX, an IBM 7090 Program to Simulate the Pattern Articulation Unit of ILLIAC III, Digital Computer Laboratory Report No. 147, July 29, 1963

☐ 4. James H. Stein, Program Description of PAX, an IBM 7090 Program to Simulate the Pattern Articulation Unit of ILLIAC III, Digital Computer Laboratory File No., in preparation

Programming (Syntactic Model)

☐ 5. R. Narasimhan, A Linguistic Approach to Pattern Recognition, Digital Computer Laboratory Report No. 121, July 10, 1962

☐ 6. R. Narasimhan, A Programming Lanaguage for the Parallel Processing of Pictures, Digital Computer Laboratory Report No. 132, January 9, 1963

☐ 7. R. Narasimhan, Syntactic Descriptions of Pictures and Gestalt Phenomena of Visual Perception, Digital Computer Laboratory Report No. 142, July 25, 1963

Programming (Bubble Chamber Automatic Scanning)

☐ 8. R. Narasimhan, A Programming System for Scanning Digitized Bubble-Chamber Negatives, Digital Computer Laboratory Report No. 139, June 24, 1963

☐ 9. R. Kevin Rice, A Preliminary Study of PAU Mcirolists in Bubble-Chamber Photographs, Digital Computer Laboratory File No. 506, February 1, 1963

☐ 10. R. Narasimhan and Brian H. Mayoh, The Structure of a Program for Scanning Bubble-Chamber Negatives, Digital Computer Laboratory File No. 507, February 7, 1963

☐ 11. Brian H. Mayoh, Bubble-Chamber Scanning Program: Syntax Table for the Compilation Phase of the Main Program, Digital Computer Laboratory File No. 538, May 28, 1963

☐ 12. R. Kevin Rice and R. Narasimhan, Bubble-Chamber Scanning Program: 1. LABEL, 2. SEARCH, Stage 1, Digital Computer Laboratory File No. 542, June 10, 1963

☐ 13. R. Narasimhan and J. P. Fornango, A Preprocessing Routine for Digitized Bubble-Chamber Pictures, Digital Computer Laboratory File No. 558, July 22, 1963

Oscilloscopic Scanner Design

☐ 14. Cyril P. Bates and B. H. McCormick, Scanner Digital Control, Digital Computer Laboratory File No., in preparation

Pattern Articulation Unit (PAU) Engineering

☐ 15. Sylvian R. Ray, The Transfer Memory, Digital Computer Laboratory Report No. 145, August 16, 1963

☐ 16. B. H. McCormick, Specifications for Thin Film, Passive Element Microcircuit Modules, Digital Computer Laboratory File No. 563, July 31, 1963

☐ 17. K. C. Smith and Dennis Hall, Observations of the Operation of a 3-Bit Bubbling Register, Digital Computer Laboratory File No. 482, August 31, 1962

☐ 18. K. C. Smith, The Pattern Articulation Unit: Circuit Design of the Iterative Array, Digital Computer Laboratory Report No., in preparation

Taxicrinic Unit (TU)

☐ 19. K. Ibuki, The Taxicrinic Unit of ILLIAC III: A Tentative Design, Digital Computer Laboratory Report No., in preparation

Arithmetic Unit (AU)

☐ 20. S. Matsushita and B. H. McCormick, Logical Design of the Main Arithmetic Unit of ILLIAC III, Digital Computer Laboratory File No., in preparation

Semiconductor Evaluation and Fast Circuit Design

☐ 21. S. Yamada, Cable Driver and Terminator Design, Digital Computer Laboratory Report No. 146, August 21, 1963

☐ 22. S. Yamada, Performance Characteristics of Fast Logic for ILLIAC III: Basic NAND and NOR Circuits, Digital Computer Laboratory Report No., in preparation

☐ 23. S. Yamada, Design of the Program-Controlled Semiconductor and Printed Circuit Board Test Console, Digital Computer Laboratory Report No., in preparation

To receive copies of any of the above listed reports, check those you would like and mail your request to:

Digital Computer Laboratory
University of Illinois
Urbana, Illinois

Those reports which are now in preparation will be sent to you as soon as they become available.

NAME ____
(Please Print)

POSITION ____

COMPLETE ADDRESS ____________________________________

____

____

DATE ____________________